

Enterprise Solutions Structure

by E. C. Plachy
P. A. Hausler

Enterprise Solutions Structure (ESS) is a major IBM initiative to establish a standard architectural framework to support creation, reuse, and maintenance of architecture and design assets for developing and delivering enterprise solutions. It draws on experiences with building customer solutions to distill "best practice" structures, models, and methods. The framework provides a rich set of architectural building blocks for solution architects and provides guidance on when and how to use this content to advantage. This paper gives an overview of ESS, describes the advantages of this approach, and serves as an introduction and context for some of the other papers in the issue.

Developing solutions for large customers, also known as enterprise customers, is becoming more complex. Factors driving this increase in complexity include:

1. Exploiting new technologies, such as Internet or intranet applications
2. Integration of solutions with a wide array of legacy systems to achieve the desired business results
3. Interoperating with multiple applications
4. Solution customization for variations in customer business processes

Further, the development and delivery of these complex systems are becoming more time consuming and risky for both IBM and its customers. Reuse of assets can help reduce the cost and risk of software development.¹ Enterprise Solutions Structure (ESS) is focused on mitigating the costs and risks of building and deploying enterprise solutions, while making

IBM's solutions and services more valuable to enterprise customers.

To serve our customers better by providing timely, adaptable, and customizable solutions and to increase our revenue and profitability on solution deployment contracts, we must focus on the creation, reuse, and maintenance of a spectrum of assets, which include methods, work products, business patterns, technology templates, and software components.

These assets need to be constructed within an overall architecture that is designed from the outset to accommodate a high rate of both business and technological change. Discipline must be applied to allow practitioners to base their solution development and delivery on standard solution architectures, methods, and tools. Enterprise Solutions Structure involves building, deploying, and evolving a set of reusable solution architectures, which, over time, can be populated with reusable software components. ESS is being used by Global Industries, IBM's solution organization, to develop solutions, and by IBM Global Services to integrate and customize solution assets for our customers, as well as to develop custom solutions.

ESS is a major initiative to establish a standard architectural framework for IBM architects and designers to develop and deliver solutions. It draws on experiences with building customer solutions to distill

©Copyright 1999 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

"best practice" structures, models, and methods that allow more effective creation and delivery of solutions. ESS is built upon the latest work-product methods used in the SI/AD (Systems Integration/Application Development) organization and consulting competencies in IBM Global Services and, where it has been possible, is expressed in terms of the newly defined Architecture Description Standard (ADS).² As such, the intellectual capital contained within can be more easily disseminated to the practitioner community of IBM Global Services.

ESS provides a rich set of architectural building blocks for solution architects, and provides guidance on when and how to use this content to advantage. As part of an architect's "tool bag," ESS packages a pre-defined, strategic set of technical reference models (also known as technology templates) with recommended vendor product mappings. Presently, ESS has five technical reference models: (1) thin-client transactional, (2) collaboration, (3) business intelligence, (4) call center, and (5) mobile computing. ESS adds significant value to IBM's e-business (electronic business) computing strategy by providing a wealth of detailed information about practical implementation scenarios, for example:

- The aggregation of business logic, data access, and services into system building blocks (*how to construct components*)
- Component topologies (*where to place components in the system*)
- Component interactions (*how to describe the overall system behavior*)
- Security management
- Systems management
- Performance characteristics

ESS provides a breakthrough for both our solutions and services businesses as we develop and deliver solutions. The ESS architectural tool bag is useful for both the development of configurable and customizable solutions for a broad set of customers and for the development of one-of-a-kind, custom solutions for specific clients. For our solutions business, ESS enables faster time-to-market through reuse of proven architectural building blocks, decreased development risk and cost, easier integration of components across solutions, and optimal use of industry domain skills for the business problems. For our services business, ESS leads to faster delivery of industry solutions by smaller teams, improves the "win rate" on bids, increases practitioners' capabilities, and minimizes the risk in solution engagements. For

our software product business, ESS enables effective integration of our products in solutions, and also identifies key requirements for product enhancements. For our customers, solutions that use ESS interoperate with and have consistent behavior with other ESS solutions, arrive earlier, are more flexible, and are more economical.

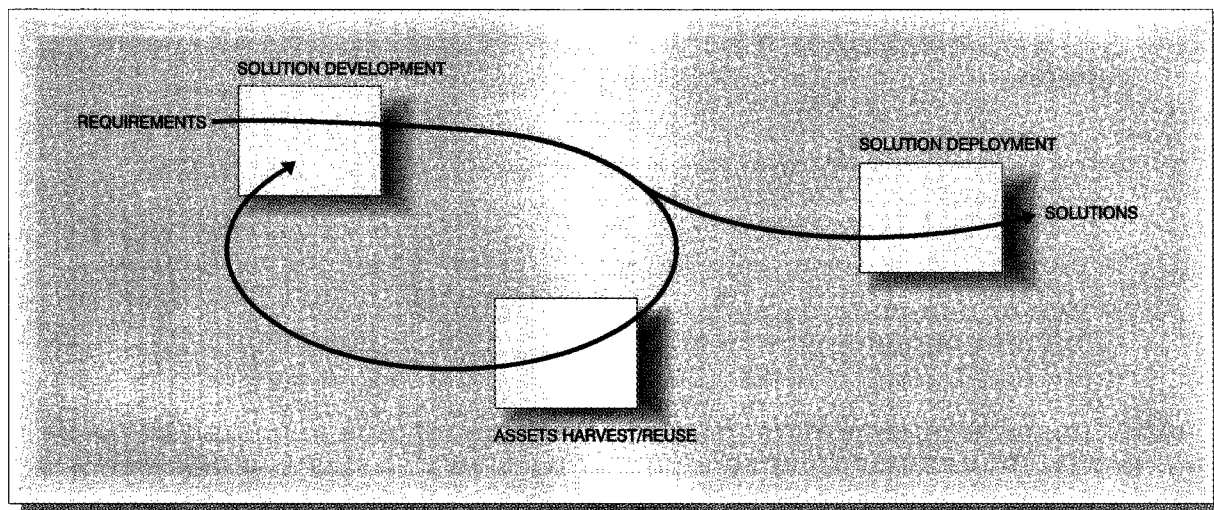
An asset-based solutions business

In late 1996, we surveyed 20 major solutions development projects in which IBM was involved, and we observed that "every project was an adventure." There was little reuse of assets, with too much custom development; tool support was inadequate. The bigger the project, the more these deficiencies mattered. For some, the problems were more technology-related. In particular, the move to the Internet as a delivery channel and a subsequent trend away from fat-client solutions threatened to render existing "stovepipe"³ applications prematurely obsolete.

What is different about developing and delivering business solutions compared to our traditional product and services businesses? To begin with, a *solution* is a combination of hardware, software products, assets, and services that together address a business problem. Like products, solutions are designed to be used by many customers, but are intended to be delivered through services engagements and to be substantially customized for each customer. A solution must provide a complete, integrated capability, including business applications and technical infrastructure, which can be integrated within an existing customer environment. Like traditional services engagements, a solution provides a capability that fits the customer's unique business and technical needs. But unlike labor-based services engagements, a solution uses an array of assets, including models, software, methods, tools, and other intellectual assets, to drive down the time and cost of delivery.

While software reuse offers the promise of decreased time to reach the marketplace, increased quality, and decreased development costs, these goals can be difficult to achieve in practice. Experience from a number of software reuse projects shows that to be successful, reuse must be applied systematically and be optimized to business goals, and three major areas must be addressed: architecture, process, and organization.¹ Enterprise Solutions Structure is focused on addressing the architecture and technical requirements for asset reuse and on the methods for creating and using assets.

Figure 1 Solution development with and without asset reuse



Asset-based solution development and deployment.

The major objective of ESS is to enable reuse of solution assets. The traditional "green field" development starts "from scratch": a set of requirements is analyzed and a solution that meets those requirements is developed and deployed. This traditional path is shown in Figure 1 by the arrow that begins at the top left (with requirements) and ends at the right (with solutions).

By contrast, asset-based solution development and deployment takes advantage of architecture and design assets by "harvesting" them from the development and deployment of each solution and reusing them with subsequent solution developments and deployments. This asset-based path is shown in Figure 1 by the arrow beginning in the middle and looping clockwise back to the top left (the solution development box). This implies that the asset-based solution development process not only has the requirements but also has the harvested assets as input, whereas the traditional custom solution development process does not have this advantage of assets as sources and thus must be developed completely from scratch each time.

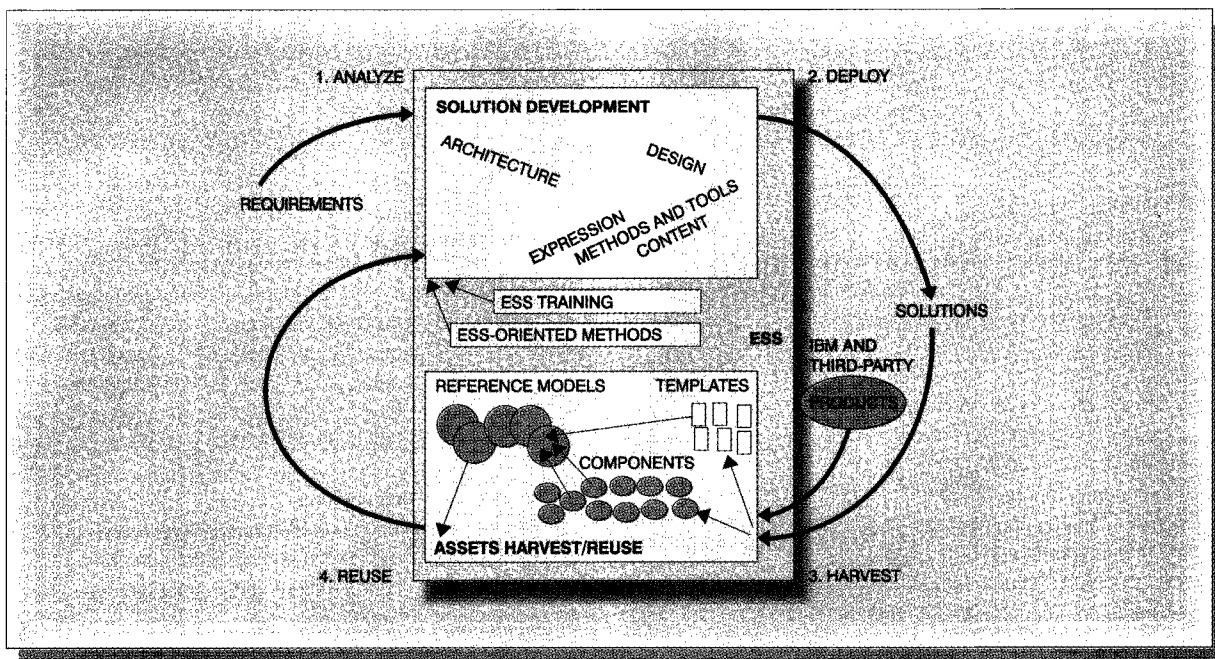
The need for a flexible solution architecture. To serve our enterprise customers better by providing timely, adaptable, and customizable solutions, while increasing our revenue and profitability on solution deploy-

ment contracts, an approach different from creating a series of disjoint and unrelated solutions is needed. Rather than misuse our creative abilities in "reinventing the wheel" on every project, we must focus on the creation, reuse, and maintenance of assets. These assets need to be constructed within an overall architecture that is designed from the outset to accommodate a high rate of both business and technological change. Discipline must be applied to "encourage" practitioners to base their solution development on a standard solution architecture, methods, and tools. Gartner Group recognizes the importance of these elements; according to Gartner Group,⁴ by the year 2000, less than 25 percent of software development organizations will have the necessary infrastructure to consistently leverage a solutions architecture. As a result, these organizations will be placed at a competitive disadvantage. In a recent report,⁵ Gartner Group asserts that 60 percent of IS (information systems) departments will find that their IT (information technology) architecture becomes obsolete faster than they can replan it, because of business and technology changes. Enterprise Solutions Structure supports building, deploying, and rapidly evolving such a set of solution architectures, which can be populated with reusable components.

ESS description

The focus of the ESS work is to capture best practices from real-life projects, to identify those ele-

Figure 2 Solution development with ESS



ments of solution architecture that are being invented repeatedly, and to collect the “proof points” and key design trade-offs. In other words, ESS has a real-world, “business problem” focus. It addresses business solution architectural issues, not just infrastructure and platform. Additionally, we are paying attention to legacy integration and to the management of business change on an ongoing basis. Our architectural patterns are designed from the outset to meet the needs of large enterprises with respect to both performance and scalability.

The architecture toolkit. The top of Figure 2 shows solution development; the bottom shows the harvesting and reuse of assets. Requirements are shown at the left; solutions are shown at the right. The outer rectangle, labeled ESS, reflects the important concept that ESS encompasses not only the asset harvesting and reuse process, but also an improved and accelerated solution development process.

The four corners of the diagram contain the four major steps that surround the many steps implied within the solution development, asset harvesting, and reuse processes:

1. Analyze (the requirements) to begin the solution development
2. Deploy (a solution) to complete the solution development
3. Harvest (assets from the solution and also from products) to begin the harvest/reuse process
4. Reuse (assets) to complete the harvest/reuse process

ESS provides a toolkit for architects. It consists of a way of describing architecture assets, based on IBM's new standard,² and a method for reusing these assets; architecture content in the form of business process patterns and technical reference models; and a tool that organizes the content.

The elements of this toolkit have been integrated into IBM's new integrated methods for asset-based application development and deployment. The major activity for the first phase of ESS has been the harvesting of best architecture practices to create a set of technical reference models and contributing to the development of IBM's new standards for describing architectures and the new asset-based methods. In future phases, ESS will concentrate on deploy-

ing and refining the technical reference models; creating work products for specifying application architecture; exploring the use of requirements to link business processes to application architecture; and developing methods for creating reusable software components and assembling customizable, configurable solutions from software components. Following is a description of the major ESS assets:

1. Architectural content

Business reference models: ESS delivers a standard approach for business modeling, including a standard language (and constructs).

Technical reference models: ESS provides and promotes a set of packaged technical reference architectures, based on best-of-breed experience, which function as the starting point for adaptation for solutions development. These define and describe, for major solution categories, the detailed componentry required along with their relationships and interactions. Design guidance is provided that shows how to choose the specific components suitable for an implementation. Currently five technical reference models are provided:

- Thin-client transactional—transacting business with customers and vendors
- Collaboration—partnering to make better business decisions
- Business intelligence—analyzing data to make decisions to improve business results
- Call center (fat-client transactional with legacy)—handling customer service interactions
- Mobile computing—disconnected or wireless connection from enterprise to remote user

2. Catalog of architectural content: ESS also provides a structured means of cataloging the architectural assets in a consistent, extensible framework using a schema based on IBM's Architecture Description Standard. This enables the use of the assets at the appropriate time in the design process, supports customization of the assets as needed, and allows new assets to be added.

3. Methods support: ESS adds value to existing methods by optimizing the reuse of business and technical reference models.

4. Product positioning and gap analysis: ESS recommends the selection of the best products for implementation of the components in the reference

models and gives criteria for selection when there is more than one product suite to choose.

5. Deployment education materials: ESS provides materials to assist skills transfer to our solutions and services community.

Other significant features. ESS places particular emphasis on the processes by which we will introduce and maintain an asset-reuse approach within the IBM solutions and services communities, rather than simply generating new assets from scratch. In the past, information assets have become buried under the weight of newer materials within a very short period of time. ESS establishes a common asset framework; a living asset repository that has extensive and effective cross referencing and indexing to support the analysis and design process, and yet is maintainable over time with direct input from the solution and services community. ESS provides the basis for building reusable assets that deliver the cross-platform benefits of IBM's e-business software products.

The ESS audience. ESS addresses the broad requirements of the IBM solutions and services business units in their goal to meet customer business needs. Particular emphasis is placed on the solution development and solution delivery aspects.

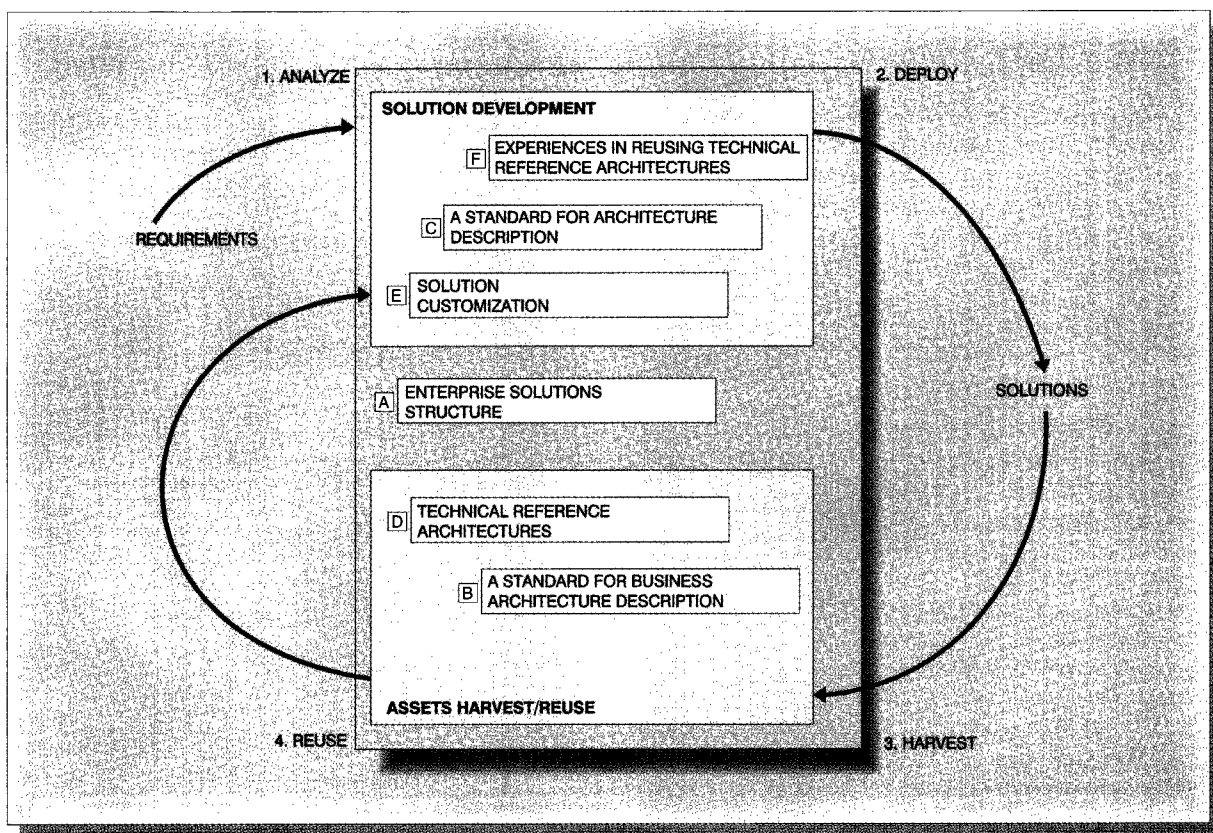
The major users of ESS are:

- Solution development architects, designing replicable business solutions
- Services architects, working on replicable solution deployment projects, custom solution development projects, and IT architecture projects for customers

The ESS papers in this issue

Some of the ESS papers in this issue describe the major elements that have been produced to date; other papers help position areas of future focus in ESS. Figure 3 helps illustrate each paper's focus within the overall ESS context. In the area of solution development, we have papers on solution customization, a standard for architecture description, and experiences in reusing technical reference architectures. In the area of asset harvesting and reuse, we have a paper on a standard for business architecture description and another on technical reference architectures.

Figure 3 Papers in this issue placed across the Enterprise Solutions Structure



Because ESS is focused on solutions, it is appropriate to start with the business architecture of solutions. McDavid's paper, "A Standard for Business Architecture Description," introduces the subject of business patterns and shows how these patterns help provide a context for an application architecture in terms of the business problem it solves, thereby improving the communication between the business domain experts and the application developers. The paper lays the groundwork for a key work in the next phase of ESS, namely articulating the variations on the business domain architecture and relating these to the set of application architectures and technology architectures.

A subject that is critical to the success of ESS is the definition of a standard for the description of architecture. Thus, it is appropriate to include the paper by Youngs et al., "A Standard for Architecture Description," in this special ESS issue. In addition to pro-

viding a standard description for our ESS technical reference models, the Architecture Description Standard also improves the communication between application developers and infrastructure specialists.

The paper "Technical Reference Architectures," by Lloyd and Galambos, describes in detail the five technical reference models (thin-client transactional, collaboration, business intelligence, call center, and mobile computing) that have been created in ESS from best practices of a number of solutions and services teams. The paper explains how an architect can easily reuse elements from these technical reference models.

The next paper, Leishman's "Solution Customization," shows where we are headed with ESS in the future to support the creation of software components to enable the rapid assembly of customizable

and configurable solutions. This will require analysis of the function that should be common, and designing the solution to accommodate the variability that will be necessary to support multiple customers.

Rounding out the ESS portion of this issue is the paper by Harris, Rothwell, and Lloyd, "Experiences in Reusing Technical Reference Architectures," which describes the encouraging results of the use of the technical reference models in actual customer engagements.

Concluding remarks

Enterprise Solutions Structure is a major initiative to establish a standard architectural framework for IBM architects and designers to develop and deliver enterprise solutions. It draws on experiences with building customer solutions to distill "best practice" structures, models, and methods that support more effective creation and delivery of solutions. ESS provides a rich set of architectural building blocks for solution architects, and guidance on when and how to use this content to advantage. ESS adds significant value to IBM's e-business computing strategy by providing a wealth of detailed information about practical implementation scenarios.

ESS provides a breakthrough for both our solutions and services businesses as we develop and deliver solutions. The ESS architectural tool bag is useful for the development of configurable and customizable solutions for a broad set of customers and for the development of one-of-a-kind, custom solutions for specific clients. For our solutions business, ESS enables faster development through reuse of proven architectural building blocks, decreased development risk and cost, easier integration of assets across solutions, and optimal use of industry domain skills for the business problems. For our services business, ESS leads to faster delivery of industry solutions by smaller teams, improves the win rate on bids, increases practitioners' capabilities, and minimizes the risk in solution engagements. For our software product business, ESS enables effective integration of our products in solutions, and also identifies key requirements for product enhancements. For our customers, solutions that exploit ESS interoperate with and have consistent behavior with other ESS solutions, are available more quickly, are more flexible, and are more economical.

Acknowledgments

The authors would like to thank Andy Patterson for his vision and leadership, especially in the early part of the project; Rock Angier for his deep, strategic thinking; and Tim Lloyd for delivering the results. We thank Deborah Leishman for her vision of how ESS should support a solutions business. We thank Burnie Blakeley for the figures he developed for the ESS papers in this issue, and we thank the reviewers, including Burnie Blakeley and John Fetvedt, for their constructive comments.

Cited references and notes

1. I. Jacobson, M. Griss, and P. Jonsson, *Software Reuse: Architecture, Process and Organization for Business Success*, Addison-Wesley Publishing Co., Reading, MA (1997).
2. R. Youngs, D. Redmond-Pyle, P. Spaas, and E. Kahan, "A Standard for Architecture Description," *IBM Systems Journal* 38, No. 1, 32-50 (1999, this issue).
3. A "stovepipe" solution has distinct parts that are not well integrated.
4. R. Hunter, "We Built the Architecture, Now How Do We Use It?" Gartner Group Strategic Planning Report SPA-480-1171 (June 7, 1995).
5. J. Furlonger, "Defining a Flexible IT Architecture," Gartner Group Research Note (April 2, 1998).

Accepted for publication November 10, 1998.

Emily C. Plachy *IBM Global Industries, 1133 Westchester Avenue, White Plains, New York 10604 (electronic mail: eplachy@us.ibm.com).* Dr. Plachy is a Senior Technical Staff Member and manager of the Solution Architecture and Reuse organization in IBM Global Industries. Her primary interests are software reuse, software development, and solution architecture. After working for Exxon Production Research Company in Houston, Texas, as a seismic applications programmer, she joined IBM's engineering/scientific compiler development team in 1982. Dr. Plachy provided the overall project management for the Parallel FORTRAN compiler, which was delivered to Cornell University as the software cornerstone of their IBM 3090 six-way NSF (National Science Foundation) Supercomputer Center. In 1990, Dr. Plachy was a senior research manager at the T. J. Watson Research Center, leading multiple application development environment research projects. In 1995, she led the development of a software architecture for interactive broadband systems. Dr. Plachy was elected to the IBM Academy of Technology in 1992. She received a B.S. degree in applied mathematics and computer science from Washington University, St. Louis, Missouri, an M.S. degree in computer science from the University of Waterloo, Waterloo, Ontario, and a D.Sc. degree in computer science from Washington University.

Philip A. Hausler *IBM Global Services, 100 East Pratt Street, Baltimore, Maryland 21202 (electronic mail: hausler@us.ibm.com).* Mr. Hausler is a senior project executive with responsibility for application and maintenance for Prudential HealthCare. Since joining IBM in 1985, Mr. Hausler has held numerous management assignments at IBM in solution and product development

Preface

The creation of software from existing resources is a well-established part of programming and software engineering for reasons of quality, productivity, and rapid development and deployment. Progress on this broad approach to reuse began at the lowest levels of programming, such as code, and has slowly reached toward the highest levels of software, such as architecture. The ability to reuse architectural assets is a vital step in the effective, efficient development of new systems and solutions from all the work that has gone before.

This issue presents the current work of the Enterprise Solutions Structure (ESS) project, a major IBM initiative focusing on the reuse of high-level software components, assets, and solutions in the large and complex systems environments of major customers. The assets being reused include architectures, methods, business patterns, and technology templates, among others. There is also a technical note on simplification of data flow diagrams. We are indebted to P. T. L. Lloyd of the IBM Object Technology Practice in Warwick, England, for his planning and coordination of this issue.

Plachy and Hausler introduce ESS through their paper on the motivation, purpose, architectural framework, development model, architectural contents, and advantages of ESS. They present the subject of high-level software reuse of solution assets, and introduce the other papers on ESS in this issue.

McDavid takes a high-level view of systems as a set of interconnected, business-oriented architectures and resulting software solutions. This leads to the description of a business systems architecture for ESS that uses key business concepts in the style of other architectures as a means for understanding and codifying systems aspects such as business requirements, boundaries of the business, and the delivery of business value.

The ESS model for reuse depends on the ability to express, in terminology and notation, the architectural aspects of existing reusable assets. The form of that expression is called the Architecture Description Standard (ADS). With ADS, assets that have been collected for reuse, or "harvested," can be systematically found and properly reused. The authors, Youngs et al., show that common descriptions of architectures in an operational model, and of components in a functional model, set the stage for active reuse.

Given the philosophy of standardization and description for reuse that surrounds ESS, Lloyd and Galambos take the next step by describing five standardized reference architectures, which are meant to cover the administrative systems needs of large businesses across the entire enterprise. These five have been carefully gleaned and abstracted from successful enterprise systems implementations.

Leishman moves beyond the basic ESS approach to describe her on-going research on solution customization. Two properties are found to be fundamental—commonality and variability—and the author describes how these properties have been applied in six examples of customizable systems. She also explains mechanisms for introducing commonality and variability, and shows in which phases of the customization life cycle the mechanisms can be applied.

Previous experience with enterprise-wide reuse has not been generally favorable. So, practical experience with such high-level and broad reuse is a necessary part of explaining the value of ESS. Harris, Rothwell, and Lloyd present the results of a number of uses of ESS in actual business situations. The conclusion is positive: ESS made it easier to rapidly and effectively provide work products that address business needs, and to support those work products later on. The authors also present the challenges they

found in further sophistication of ESS and the ESS approach.

In a technical note, Millet presents a modification to usual data flow diagram notation and use in which each data store symbol represents a whole database instead of a table. The claim is that such a modification makes it easier to create, understand, and maintain data flow diagrams.

As the *Journal* begins its 38th year, we would like to acknowledge the support of readers, authors, and referees that makes such a long history possible. We thank you and encourage you to continue your interest and participation in this publication. It also seems appropriate at such a time to state a few facts that sometimes escape us as we focus on a single paper, theme, or special issue. First, this quarterly publication is a refereed technical journal, which means that the integrity of each paper is ensured by a process that depends upon peer reviews of content, currency, and value by recognized experts within and outside IBM. Second, it is intended for the software and systems professional and applied research community worldwide. The papers are written for a technically aware readership, and we welcome submissions by knowledgeable authors around the globe, within and outside IBM. Third, the *Journal* has over 50 000 subscribers worldwide. Of those, approximately two-thirds are technical professionals and researchers outside IBM and one-third are IBM employees; two-thirds are in the United States and one-third are outside the United States.

The next issue of the *Journal* will be a unique, retrospective issue on some of the key turning points in the evolution of computing during the twentieth century, as reflected in the pages of the *Journal* since 1962. There will also be commentary on the importance of those turning points and the related papers by key individuals who saw it happen, written from their perspective at the close of the century.

Gene F. Hoffnagle
Editor